

Pre-Hardware Optimization of Spacecraft Image Processing Software Algorithms and Hardware Implementation

Semion Kizhner, David J. Petrick, Thomas P. Flatley, Phyllis Hestnes, Marit Jentoft-Nilsen
National Aeronautics and Space Administration

Abstract

Spacecraft telemetry rates and telemetry product complexity have steadily increased over the last decade presenting a problem for real-time processing by ground facilities. This paper proposes a solution to a related problem for the Geostationary Operational Environmental Spacecraft (GOES-8) image data processing and color picture generation application. Although large super-computer facilities are the obvious heritage solution, they are very costly, making it imperative to seek a feasible alternative engineering solution at a fraction of the cost. The proposed solution is based on a Personal Computer (PC) platform and synergy of optimized software algorithms, and re-configurable computing hardware (RC) technologies, such as Field Programmable Gate Arrays (FPGA) and Digital Signal Processors (DSP). It has been shown in [1] and [2] that this approach can provide superior inexpensive performance for a chosen application on the ground station or on-board a spacecraft. However, since RC technologies are still maturing, intensive pre-hardware steps are necessary to achieve the benefits of hardware implementation. This paper describes these steps for the GOES-8 application - a software project originally developed using the Interactive Data Language (IDL) (Trademark of Research Systems, Inc.) on a Workstation/UNIX platform. The solution involves converting the application to a PC/Windows/RC platform, selected mainly by the availability of low cost, adaptable high-speed RC hardware. In order for the application to run on this PC/RC hybrid system, the IDL software was modified to account for platform differences. It was interesting to examine the gains and losses in performance on the new platform, as well as unexpected observations before implementing RC hardware. After substantial pre-hardware optimization steps in

the PC environment, the necessity of RC hardware implementation for bottleneck code became more evident. The problem was solved beginning with the methodology described in [1], [2], and implementing a novel methodology for this specific application. The PC-RC interface bandwidth problem for the class of applications with moderate input-output data rates but large intermediate multi-thread data streams has been addressed and mitigated. This opens a new class of satellite image processing applications for bottleneck problems solution using RC technologies. The issue of a science algorithm level of abstraction necessary for RC hardware implementation is also described. Selected Matlab functions already implemented in hardware were investigated for their direct applicability to the GOES-8 application with the intent to create a library of Matlab and IDL RC functions for ongoing work. A complete class of spacecraft image processing applications development using re-configurable computing technology to meet real-time requirements, including methodology, performance results and comparison with the existing system, is described in this paper.

Table of Context

Abstract

Introduction

- 1.0 Problem engineering complexity and solution methodology
- 2.0 Heritage application and its operations overview
- 3.0 Porting the UNIX/IDL application to the PC/Windows platform, modifying code, compilation and run
- 4.0 Application performance timing methodology
- 5.0 Timing tests on the PC/Windows98 platform
- 6.0 Algorithm definition, optimization and RC hardware implementation